

게임 및 가상현실에서의 특수효과를 위한 입자 시스템 에디터

김 응 곤*, 송 승 현

순천대학교 정보통신공학부*, 순천대학교 대학원 컴퓨터과학과

Particle System Editor for Special Effects in Game and Virtual Reality

Eung-kon Kim*, Seung-heon Song

Computer & Communication Engineering, Suncheon National University*

Graduate School of Computer Science, Suncheon National University

E-mail : kek@sunchon.ac.kr* abulapia@sunchon.ac.kr

요 약

게임과 영화는 현재 엔터테인먼트 산업에서 가장 각광을 받고 있는 분야이며, 이러한 분야에서는 입자 시스템을 이용하여 불꽃, 폭발, 연기, 액체, 눈, 비, 먼지와 같은 특수효과를 만들어 낸다.

게임 및 가상현실에서 상위 수준의 그래픽 라이브러리인 입자 시스템 API를 사용하면 위와 같은 특수효과를 사실적으로 표현할 수 있게 한다. 입자 시스템 API를 적용 시 개발자가 원하는 형태의 특수효과가 구현될 때까지 파라미터 값을 계속 바꿔가며 소스코드를 컴파일하여야 하며, 각 파라미터들 간의 연관성있는 세밀한 조정이 이루어지기까지 많은 시간이 필요하다는 단점을 가지고 있다.

이에, 본 논문에서는 온라인 게임 및 실시간 가상현실에 실제 적용할 수 있는 입자 시스템 API를 개발하고 위치, 속도, 색상, 투명도, 크기, 수명, 2차 위치, 2차 속도 등의 속성 조절을 통해 손쉽게 특수효과를 생성할 수 있는 입자 시스템 에디터를 개발하고자 한다.

ABSTRACT

In games and films, the most highlighted fields in entertainment industry, those special effects such as flame, explosion, smoke, liquid, snow, rain and dust are generated through the particle system.

Special effects can be expressed realistically by the particle system API that is a graphic library of high level in game and virtual reality. When developers apply the particle system API in applications, they must exchange parameters repeatedly and compile source codes until special effects that they want are expressed, and It takes much time until the minute control that have relations between each parameters.

This paper develops a particle system API usable in on-line game and real-time virtual reality and presents particle system editor that can see and create special effect easily through attributes adjustment such as position, velocity, color, transparency, size, age, the secondary position, the second velocity etc.

키워드

Particle System, API, Special Effect, Game, Virtual Reality

1. 서 론

게임과 영화는 현재 엔터테인먼트 산업에서 가장 각광을 받고 있는 분야이며, 이러한 분야에서는 입자 시스템을 이용하여 불꽃, 폭발, 연기, 액

체, 눈, 비, 먼지와 같은 특수효과를 만들어 낸다.

게임 및 가상현실에서 상위 수준의 그래픽 라이브러리인 입자 시스템 API를 사용하면 위와 같은 특수효과를 사실적으로 표현할 수 있으며, 영상 연출을 위한 개발자는 특수효과를 위한 객체 단위의 프로그래밍 작업이 가능하게 되므로 기본

개념의 이해를 통해 비교적 쉽게 여러 가지 특수 효과를 필요로 하는 온라인 게임이나 가상현실 프로그램에 적용할 수 있다.

하지만 이러한 방식으로 입자 시스템 API를 적용할 경우, 개발자가 원하는 형태의 특수효과가 구현될 때까지 파라미터 값을 계속 바꿔가며 소스코드를 컴파일하여야 할 뿐만 아니라, 각 파라미터들 간의 연관성있는 세밀한 조정이 이루어지기까지 많은 시간이 필요하다는 단점을 가지고 있다.

이에, 본 논문에서는 온라인 게임 및 실시간 가상현실에 실제 적용할 수 있는 입자 시스템 API를 개발하고 위치, 속도, 색상, 투명도, 크기, 수명, 2차 위치, 2차 속도 등의 속성 조절을 GUI환경을 통해 손쉽게 특수효과를 생성할 수 있는 입자 시스템 에디터를 개발하고자 한다.

1장 서론에 이어 2장 관련연구에서는 게임 및 가상현실을 위한 입자 시스템 API에 대해 기술한 후, 3장에서 구현하고자하는 입자 시스템 에디터에 대해 설계하고, 4장에서는 입자 시스템 API의 실제 적용을 위한 입자 시스템 에디터를 구현한다. 5장에서 결론을 내린다.

II. 입자 시스템 API

OpenGL 그래픽 라이브러리에 기반한 입자 시스템은 특수효과 발생을 위한 상위 수준의 그래픽 API로 구성되기 때문에 유체속성, 불, 불꽃 속성, 기상현상 등의 여러 가지 특수효과에 대해 설정값의 변경으로 특수효과를 표현할 수 있으므로, 게임이나 가상현실 등의 응용 프로그램에서 이를 변형하여 사용할 수 있다는 장점을 가지고 있다.

다음 그림 1은 OpenGL에 기반한 입자 시스템을 특수효과 라이브러리로 적용하였을 때 Game과 가상현실 프로그램 등에서 어떻게 확장되고 응용될 수 있는지 보여준다.

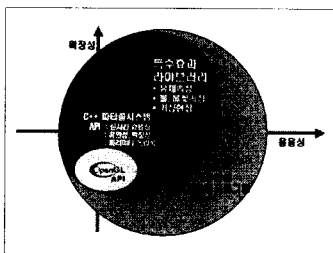


그림 1. 입자 시스템을 사용한 특수효과 라이브러리의 확장 및 응용

본 논문에서는 게임이나 가상현실 응용 프로그램의 실시간 효율성과 프로그래머가 손쉽게 다양

한 효과를 만들어낼 수 있는 유연성을 가지고 있으며, 개개의 파라미터에 대해 독립성을 유지하고 확장성이 있는 상위 수준의 입자 시스템에 대해 구축된 입자 시스템 API를 기반으로 여러 가지 특수효과의 구현 및 시각화를 손쉽게 하며, 생성된 효과를 다양히 적용시키기 위한 입자 시스템 에디터를 구현하고자 한다.

III. 입자 시스템 에디터의 설계

3.1 입자 시스템 API와 OpenGL

입자 시스템 API은 OpenGL에 기반하여 만들어진 API의 성격을 가지고 있지만 특수효과를 위한 파티클의 생성과 설정, 함수값의 설정은 OpenGL과 독립된 모습을 보여주고 있다. OpenGL은 입자 시스템 API의 설정 외에 생성되는 가상세계를 초기화 시켜주며, 장면이 표현되기 위한 윈도우 및 오브젝트와 3차원 가상공간을 정의하는 역할을 하고, 입자 시스템 API은 입자들의 생성 및 설정을 담당하게 된다.

3.2 윈도우와 입자 시스템 에디터

OpenGL을 기반으로 입자 시스템 API를 적용하여 만들어진 입자 시스템 에디터는 윈도우 기반 응용프로그램으로서 동작을 하게 되는데, 이때 화면에 출력되는 입자 시스템의 그래픽 출력은 그래픽 디바이스 인터페이스와 디바이스 컨텍스트에 의해 이루어진다.

그래픽 디바이스 인터페이스 즉, GDI는 윈도우가 하드웨어를 제어할 수 있도록 응용 프로그램에 제공하는 모든 기능을 일컫는데, 응용 프로그램에 있어서 장치 독립적인 그래픽 동작을 수행하게 되므로, 실제 그래픽 객체들의 속성과 그래픽 모드를 정의하는 자료의 집합체로 출력장치에 내용을 표시하는데 필요한 정보를 가진 자료구조인 디바이스 컨텍스트(DC)는 윈도우 화면에 출력되는 모든 내용을 출력하게 된다.

다음 그림 2는 윈도우 상에서 동작하는 입자 시스템 에디터의 개념도를 보여준다.

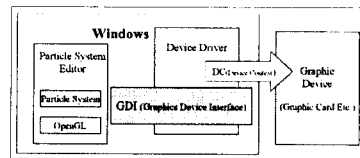


그림 2. 윈도우 상에서 입자 시스템 에디터의 동작

3.3 입자 시스템 에디터의 구성

입자 시스템 에디터의 구성은 크게 4부분으로 나누어진다. 이는 입자 속성 모듈, 입자 함수 모

들, 속성과 영역 모듈, 그리고 그래픽 유저 인터페이스의 구현을 위한 GUI 모듈이다.

다음 그림 3은 입자 시스템 에디터를 각 기능별로 모듈화한 구성도이다.

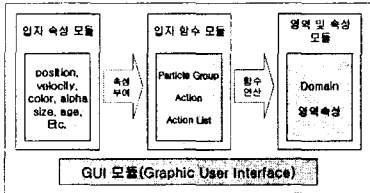


그림 3. 입자 시스템 에디터의 모듈 구성도

3.3.1 입자 속성 모듈

입자 시스템 에디터 내의 입자는 위치(position), 속도(velocity), 색상(color), 알파(alpha), 크기(size), 수명(age), 2차 위치(secondary position), 2차 속도(secondary velocity) 등의 속성으로 구성된 모듈에 의해 다루어진다.

3.3.2 입자 함수 모듈

에디터는 입자 그룹함수, 액션함수, 액션리스트 함수 등 3가지 종류의 함수로 구성한다. 개발된 에디터 내 입자 시스템 API 함수 이름은 pFunctionName 형태를 취하게 되며, 사용되는 입자 시스템 API 함수는 다음과 같이 고려된 함수를 사용한다.

· 입자 그룹 함수

모든 입자들은 같은 힘에 의해 작용하는 입자들의 집합인 입자 그룹 내에 존재하며, 모든 입자 시스템 API 함수들은 현재의 입자 그룹에만 적용되고 모든 액션들은 현재의 입자 그룹 내의 모든 입자들에 적용되도록 한다. 입자 그룹들은 최초에 비어있고 pSource 함수의 호출에 의해 그룹 내에 새로운 입자들을 만들어 내도록 하여 입자 그룹이 최대 크기에 도달하면 더 이상 입자의 추가가 불가능하게 한다.

· 액션 함수

액션들은 현재의 입자 그룹 내의 입자들의 속성을 변경시키는 함수들이며, 많은 액션함수들은 입자가 어떤 영향권에서 멀어짐에 따라 효과를 감소시키는 감쇠함수를 필요로 하는데 본 모듈에서는 입자 시스템의 동질성을 위하여 감쇠함수를 필요로 하는 액션함수들에 대하여 다음과 같은 감쇠함수를 사용하였다.

$$f_{m,r}(r) = \begin{cases} 0 & r \geq r_{max} \\ \frac{m}{r^2 + \epsilon} & r < r_{max} \end{cases}$$

여기서 m은 입자의 크기, r은 거리(반경), rmax는 최대 거리, ε은 거리 r이 0에 가까워짐에 따라 f(r)이 무한대가 되지 않도록 하는 값이다. 다음 그림은 입자의 크기가 1이고, ε의 값이 1인 이 감쇠함수의 그래프를 나타낸다. 이 함수를 이용함으로써 입자의 크기와 ε만으로써 더 예민한 감쇠효과를 낼 수 있으며 실행 시 효율성을 크게 증가시키게 된다.

· 액션 리스트 함수

특수효과를 만들어 내는데 필요한 모든 연산들을 캡슐화하기 위하여 필요한 액션들을 모아 놓은 액션 리스트를 사용한다. 액션 리스트는 에디터에 특수효과의 추상화를 제공할 뿐 아니라 어떠한 그래픽스 하드웨어에서도 프로그래밍하기 쉬운 인터페이스를 제공해 준다.

3.3.3 속성과 영역

영역 즉 도메인은 어떤 액션이나 상태함수에 파라미터로 사용되며, 입자 생성 시에 도메인 내에서 임의로 선택된 위치에서 입자를 만들어낸다. 본 연구에서는 구, 평면, 육면체, 실린더, 원뿔, 원반, 삼각형, 직사각형, 선분, 점 등 다양한 형태의 영역을 사용하도록 하였다.

다음 그림 4는 원반 영역 내에서 임의로 선택된 속도 벡터를 나타낸다.

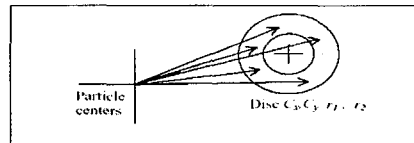


그림 4. 원반영역 내에 임의로 선택된 속도벡터

3.3.4 GUI 모듈

입자 시스템 에디터의 파라미터 값을 사용자 임의로 조작하여 시각화되려면 필요한 파라미터를 입력하기 위한 각각의 필드가 제공되어야 한다.

이때, 입자 시스템 에디터 각각의 필드는 다양한 효과의 생성을 위해 입자 속성 모듈의 파라미터인 색상, 투명도, 크기, 수명, 1차 위치와 속도, 2차 위치와 속도 등을 수치입력과 슬라이드 바 형태로 조절할 수 있도록 한다.

IV. 입자 시스템 에디터의 구현

입자 시스템 에디터는 파라미터 입력을 용이하게 하기 위해 MFC를 이용하여 GUI 기반의 인터페이스를 구축하였으며, 에디터의 오른쪽에 주요 도메인의 선택과 파라미터 입력을 위한 슬라이더

와 입력 창을 생성하고 리셋 버튼과 시물레이션 버튼으로 화면을 갱신, 생성할 수 있게 하였다.

그리고 입자 시스템 API의 주요 소스코드와 윈도우 리소스 간의 연결은 Win32 API에 기반으로 코딩하였다.

다음 그림 5는 입자 시스템 에디터의 모습을 나타낸다.

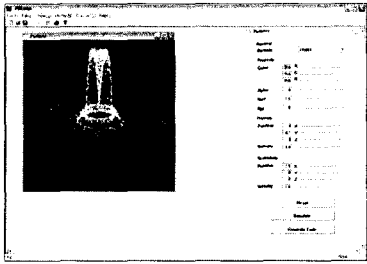


그림 5. 특수효과 생성을 위한 입자 시스템 에디터

본 입자 시스템 에디터의 구현 환경은 다음 표 1과 같다.

표 1. 구현 환경

소프트웨어	하드웨어
개발환경 : OpenGL, Visual C++	Graphic Card : Geforce FX5600
운영체제 : Windows XP Professional	CPU : Pentium4 2.6Ghz

다음 그림 6는 입자 시스템 에디터의 파라미터 값 변경에 의해 생성된 소스코드를 보여준다.

```

:
void Waterfall(bool first_time = true)
{pVelocityD(PDBlob, 0.1, 0, 0.1, 0.004);
pColorD(1.0, PDLine, 0.8, 0.9, 1.0, 1.0, 1.0, 1.0);
pSize(1.5);
pStartingAge(0);

if(first_time)
{action_handle = pGenActionLists(1);
pNewActionList(action_handle);
pCopyVertexB(false, true);
pSource(50, PDPoint, -4, 0, 6);
pGravity(0.0, 0.0, -0.01);
pKillOld(250);
pBounce(0, 0.01, 0, PDSphere, -1, 0, 4, 1);
pBounce(0, 0.01, 0, PDSphere, -2.5, 0, 2, 1);
pBounce(0, 0.01, 0, PDSphere, 0.7, -0.5, 2, 1);
pBounce(-0.01, 0.35, 0, PDPlane, 0,0,0, 0,0,1);
pMove();
If(first_time)
pEndActionList(); }
:
    
```

그림 6. 입자 시스템 에디터에 의해 생성된 소스코드

V. 결론 및 향후과제

본 연구에서는 온라인 게임이나 실시간 가상현실에 손쉽게 적용할 수 있는 특수효과를 효율적으로 생성할 수 있는 에디터를 입자 시스템 API에 기반하여 설계, 구현하였다.

입자 시스템 API는 실시간 동적 입자 시스템으로서 게임이나 가상현실에서 불, 불꽃, 비, 눈 등의 여러 가지 특수효과를 적용하는데 있어 기존의 개별적인 알고리즘 보다 활용성과 유연성 등이 뛰어나 그 개발 시간과 비용을 크게 줄인다는 장점이 있지만, 개발자가 시각적으로 만족할만한 특수효과가 구현될 때 까지 파라미터를 바꿔가며 소스코드를 작성하여야 할 뿐만 아니라, 다양한 효과를 위한 각 파라미터들 간의 연관성있는 세밀한 조정을 위해 많은 시간을 투자하여야 한다는 단점이 있다.

본 논문에서는 입자 시스템을 시각화하는데 가장 중요한 요소인 입자의 위치, 속도, 색상, 투명도, 크기, 수명 등 여러 가지 속성을 조절하여 개발자가 원하는 특수효과를 자유롭게 생성할 수 있도록 그래픽 유저 인터페이스를 사용하여 입자 시스템 에디터를 구현하였다.

입자 시스템 에디터의 광범위한 활용은 특수효과 라이브러리의 구축을 용이하다는 장점을 가지고 있으므로 온라인 게임이나 실시간 가상현실에서 사용자의 흥미를 크게 증진시키는 역할을 수행하게 될 것으로 기대된다.

참고문헌

- [1] Reeves, W. T. "ParticleSystems - A Technique for Modeling A Class of Fuzzy Objects". Proc. of SIGGRAPH '83, Detroit, Michigan, July, 1983.
- [2] McAllister, D. K. "The Design of an API for Particle Systems" <http://cs.unc.edu/~davemc/Particle>, 1999
- [3] Leech, J. P. and R. M. Taylor. "Intractive Modeling Using Particle Systems" . Proc of 2nd Conference on Discrete Element Methods, MIT, 1993
- [4] Allen, M. B. Flow - a particle animation application. <http://www.dnai.com/~mba/software/flow/>, 1999
- [5] <http://www.opengl.org>
- [6] Neider, J. T. Davis, et al. OpenGL Programming Guide. Adison Wesley, 1993
- [7] 송승헌, 김용곤, 파티클 시스템을 이용한 게임 및 가상현실에서의 특수효과, p94-102, 7(4), 멀티미디어학회지, 2003.
- [8] 김세준, 송승헌, 김용곤, 파티클 시스템 API

를 이용한 특수효과 구현, p609-612, 6(2),
멀티미디어학회 추계학술발표회, 2003.

감사의 글
본 연구는 2003년도 정보통신부에서 지원하는
기초기술연구지원사업으로 수행되었음.